

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **07160905 A**

(43) Date of publication of application: **23.06.95**

(51) Int. Cl

**G06T 15/00**

(21) Application number: **05307719**

(22) Date of filing: **08.12.93**

(71) Applicant: **HITACHI LTD**

(72) Inventor:  
**TOKUYAMA HIDEKI  
SAKAHARA TOORU  
KUWANA TOSHIYUKI**

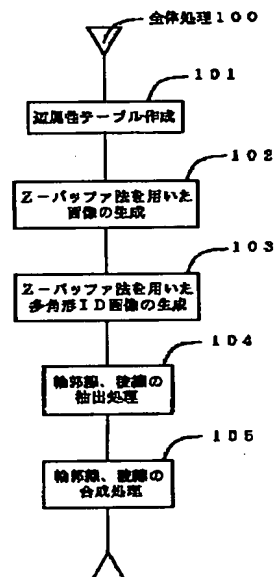
(54) **METHOD AND DEVICE FOR PLOTTING  
CONTOUR LINE AND RIDGE LINE**

COPYRIGHT: (C)1995,JPO

(57) Abstract:

**PURPOSE:** To obtain the method for generating a picture whose object shape is easily recognized by detecting a contour line and a ridge line of a display object without error and adding a contour line and a ridge line to the object so as not to be thickened.

**CONSTITUTION:** A side attribute table required to detect a contour line and a ridge line from a 3-dimension shape data given at first is generated in the step 101. Then a picture whose hidden face is deleted is generated in the step 102 and a polygonal ID picture is generated in the step 103. Then in the step 104, the side attribute table generated in the step 101 is collated with the polygonal ID picture generated in the step 103 to discriminate whether each picture element forms a contour line or a ridge line. Then the contour line and the ridge line are synthesized to the picture whose hidden side is deleted in the step 105, then a picture whose object shape is easily recognized is generated. Furthermore, a wire frame picture whose hidden line is deleted is generated by displaying only the detected contour line and ridge line.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-160905

(43) 公開日 平成7年(1995)6月23日

(51) Int.Cl.<sup>6</sup>

G 0 6 T 15/00

識別記号

庁内整理番号

F I

技術表示箇所

9365-5L

G 0 6 F 15/ 72

4 5 0 A

審査請求 未請求 請求項の数 8 O L (全 7 頁)

(21) 出願番号

特願平5-307719

(22) 出願日

平成5年(1993)12月8日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 徳山 秀樹

神奈川県横浜市戸塚区吉田町292番地株式

会社日立製作所システム開発研究所内

(72) 発明者 酒井原 徹

神奈川県横浜市戸塚区吉田町292番地株式

会社日立製作所システム開発研究所内

(72) 発明者 桑名 利幸

茨城県日立市大みか町五丁目2番1号株式

会社日立製作所大みか工場内

(74) 代理人 弁理士 小川 勝男

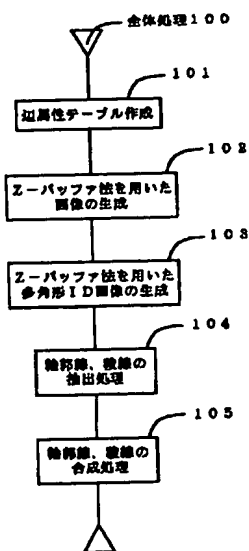
(54) 【発明の名称】 輪郭線および稜線の描画方法および描画装置

(57) 【要約】

【目的】 計算機による3次元図形表示において、物体形状をわかりやすく、かつ物体形状を損なうことなく表示するために、輪郭線および稜線を誤りなく検出し、表示図形に付加する方法およびその装置を提供すること。

【構成】 多面体近似で与えられる3次元物体形状から、CPUにより多角形と多角形との接続具合の情報を作成し、同時に法線ベクトルを基にその辺の種別を判定し、辺に対応させて記憶する。また、表示画面の各画素にどの多角形が表示されているかを記憶しておく。そして、辺をたどり、表示されている多角形が辺の両側で接続具合を保っていない場合を輪郭線として検出、描画する。また、接続具合を保っている場合には、辺の種別が稜線の場合のみ稜線として検出、描画する。

図 1



## 【特許請求の範囲】

【請求項1】多角形近似データで与えられた3次元物体を2次元画像として表示する場合において、与えられた多角形の接続具合の情報、すなわち、多面体近似データ上で、どの多角形とどの多角形とが互いに接しているかの情報を得、図形表示がその接続具合を保っていないところを輪郭線として検出し、輪郭線として他の部分とは異なった色で表示する方法。

【請求項2】請求項1の輪郭線を検出する方法において、与えられた全ての多角形の全ての辺をたどり多角形と多角形との接続具合の情報を多角形の辺に対応させて記憶し、また、画面の各画素に表示されている多角形の識別子を記憶した画像を生成し、辺をたどり、今たどっている画素の周囲画素に記憶された多角形の識別子が、接続具合を保っているかどうかを判定し、保っていない部分を輪郭線とする方法。

【請求項3】請求項2の多角形の識別子を記憶した画像の接続具合を判定する方法において、この接続具合を画素単位で判定し、判定する画素の周囲8画素に接続した2つの多角形が両方とも1画素以上表示されていれば接続具合を保っているとし、どちらか一方の多角形のみが1画素以上表示されていれば保っていないとする判定方法。

【請求項4】多角形近似データで与えられた3次元物体を2次元画像として表示する場合において、多角形間の接線がもともと角張っている線なのか近似によって現れる線なのかを分類し、もともと角張っている線のみを稜線として検出し、稜線として他の部分とは異なった色で表示する方法。

【請求項5】請求項4の多角形間の接線の分類法において、多角形と多角形との接線上で2方向の法線が存在する場合を稜線、法線が1方向のみ存在する場合を近似によって現れる線と分類する方法。

【請求項6】請求項2の輪郭線を検出、表示する方法と、請求項4の稜線を検出、表示する方法の両者を同時に実現する方法。

【請求項7】多面体近似データで与えられた3次元物体を2次元画像として表示するための図形生成装置および表示装置であって、図形データを描画命令に変換する手段と、前記描画命令を受けて表示画面を生成するグラフィックスプロセッサと、これに命令を送る手段と、表示するための色情報を記憶するフレームメモリと、前記フレームメモリの色情報を表示するための手段と表示多角形の多角形識別子を表示画面の各画素ごとに記憶するためのメモリと、与えられた多面体近似データから多角形の接続具合の情報を多角形の辺に対応させて生成する手段と、多角形の接続具合の情報と多角形IDを記憶したメモリとを照合し輪郭線または稜線を検出し、これら輪郭線または稜線を他の部分とは異なった色で表示画像に付加する手段と、を有することを特徴とする図形生成装

置。

【請求項8】請求項7の輪郭線または稜線を検出し、これら輪郭線または稜線を他の部分とは異なった色で表示画像に付加する手段において、直線の画素を発生する手段と、発生した画素の周囲画素の多角形IDを読み込む手段と、前記多角形IDが多角形の接続具合の情報を満足しているかどうかを判定する事により輪郭線または稜線であるかを判定、検出する手段と、前記輪郭線または稜線を他の部分とは異なった色で表示画像に付加する手段と、を有することを特徴とする装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、計算機を用いた図形表示法に係り、特に物体形状を理解しやすく表示するための輪郭線および稜線の描画方法およびその装置に関する。

## 【0002】

【従来の技術】従来より、計算機によって3次元物体形状を2次元の画像として表示する場合、物体形状は多角形近似したデータで与えられ、この与えられた多角形の輪郭を線で表示する方法（線画による表示法）や、多角形内部を濃淡や色を変えて塗りつぶして描画する方法（面画による表示法）などがとられてきた。

【0003】写実的な画像を生成することが目的の場合、光源の位置や物体表面の反射率などを基に、物体表面上の各点の輝度を計算し、陰影付けされた画面として表示する方法が通常用いられている。代表的な方法としては、Z-バッファ法（電子情報通信学会 編、ニューメディア技術シリーズ コンピュータグラフィックス、P121-P122）やレイトレーシング法（日本図学会、CGハンドブック、P417-P418参照）などがあげられる。

【0004】また、Z-バッファ法などによって生成された画像においては、光源の位置や反射率等によって、物体の形状や、複数の物体が並んでいるときの配置状況等が認識し難くなる場合がある。これを解消するためには、輪郭線および稜線の強調が効果的であることが従来から知られており、計算機によって、輪郭線および稜線を自動的に付加する方法が提案されてきた。すなわち、特公平3-123985号公報記載のように、Z-バッファ法によって生成された距離画像を基に、輪郭線および稜線を検出し、輪郭線および稜線を付加し表示する方法が提案されている。ここで、輪郭線とは、ある物体の見える部分と見えていない部分との境界線を、稜線とは輪郭線以外の物体の角ばった部分を表す内形線をそれぞれ指すものとする。

## 【0005】

【発明が解決しようとする課題】上記従来技術の距離画像を基にした輪郭線および稜線を検出する方法には、以下の問題点が存在する。

【0006】(1)距離画像の1次微分値および2次微分

値をとり、これらの値がピーク値をとるところを輪郭線および稜線として検出している。しかし、この距離画像は通常整数値にしているために量子化誤差を含んでいる。例えば、画面に平行に近い平面の距離画像は、ある部分でステップ状の変化を生じる。このようにステップ状に変化している部分では微分値がピーク値をとり、輪郭線または稜線として検出されるので、検出に誤りが生じてしまう。

【0007】(2)輪郭線が物体境界の内側と外側の両側で検出されるために、表示物体の形状や大きさによらずに太くなる。このために、例えば輪郭線の色を背景と同色にすると、表示物体が一回り小さく見え、また輪郭線の色が背景と大きく異なる(例えば輪郭線が黒で背景が白の場合など)と、人によっては、物体の境界を線の外側または内側と見てしまい、一回り大きくまたは小さく見てしまう。このことは、物体が小さく表示される程、表示物体の面積に占める輪郭線および稜線の割合が大きくなるため、顕著に現れる。

【0008】本発明の目的は、上記の問題を解決し、表示物体の輪郭線および稜線を誤りなく検出し、また、太くならないように輪郭線および稜線を付加することにより、物体形状がわかりやすい画像を生成する方法を提供することにある。

【0009】

【課題を解決するための手段】上記目的を達成するために、本発明では、多面体近似で与えられる3次元物体形状データから、多角形と多角形との連続性すなわち、どの多角形とどの多角形とが互いに接しているかを調べ、この連続性のデータを多角形の辺に対応させると同時に、この辺が角張った辺(稜線)なのか、多面体近似によって平面または曲面が分割された線(以後近似線と呼ぶ)なのかも判定し、これらを辺属性テーブルとして記憶しておく。そして、これらの辺を表示する際に、表示画像上においてこの辺の両脇で、多角形の連続性を満足していない辺が輪郭線となる。また、連続性を満足している場合には、稜線か近似線なのかは先に判定済みなので、稜線も検出できる。そしてまた、辺の描画を1画素幅とすれば、線幅は太くならない。これにより先の目的を達成している。

【0010】

【作用】本発明ではまず、前記辺属性テーブルを作成し、通常の表示画像および色の代りに多角形の識別子を記憶した画像すなわち、多角形ID画像を作成する。そして、辺属性テーブルと多角形ID画像とを照合することにより、多角形の連続性を判定し、不連続ならば輪郭線と判定し、輪郭線として通常の表示画像に他の部分とは異なった色で付加する。また、連続である場合には稜線であるか否をみて、稜線ならば稜線として通常の表示画像に他の部分とは異なった色で付加する。このようにして物体の輪郭線、稜線を誤りなく検出し、付加するこ

とができ、線幅も1画素幅で表示することができ物体形状も損なうことはない。

【0011】

【実施例】以下本発明の実施例を図1～図11により説明する。

【0012】図1に本発明の全体の処理の流れを示す流れ図を示す。初めに、全体処理100の概要を示し、詳細については後に説明することにする。まず初めに、与えられた3次元形状データから、ステップ101により、輪郭線、稜線を検出するために必要な辺属性テーブルを作成する。次に、ステップ102により隠面消去された画像を生成する。次に、ステップ103によって多角形ID画像を作成する。そして、ステップ104によって、ステップ101で作成された辺属性テーブルとステップ103で作成された多角形ID画像と照合することにより、各画素が輪郭線か稜線かが判定できる。そしてステップ105によって、これら輪郭線、稜線を、先の隠面消去された画像に合成することにより、物体形状がわかりやすい画像が生成される。また、検出した輪郭線、稜線のみを表示することにより隠線消去されたワイヤフレーム画像も生成できる。

【0013】次に、本発明を組み込んだシステムの一実施例について、図10を用いて説明する。このシステムは、CPU1001、主メモリ1002、グラフィックスプロセッサ1005、Zバッファ1008、フレームメモリ1009、多角形IDメモリ1010、表示装置1011、システムバス1012からなる。

【0014】ここで、CPU1001は、グラフィックスプロセッサ1005への描画命令を出したり、辺属性テーブルの作成(ステップ101)等の処理を行い、主メモリ1002は表示のためのプログラム、図形データの保存エリア1003、辺属性テーブルエリア1004を含む。グラフィックスプロセッサ1005は、CPU1001から受けた描画命令を実行し、フレームメモリ1009および多角形IDメモリ1010に画像を生成し、通常描画部1006と輪郭線稜線発生部1007を含む。Zバッファ1008は隠面消去を実行するために必要なデータを記憶し、フレームメモリ1009は図形を表示するための色情報を記憶し、多角形IDメモリ1010は輪郭線、稜線を検出するために必要な多角形ID画像を記憶し、表示装置1011は、フレームメモリ1009上の画像を表示する。

【0015】以下に、全体処理100の各ステップの詳細を説明する。

【0016】ステップ101の説明：次に辺属性テーブルの作成ステップ101について図2、図4～図7を用いて説明する。

【0017】図2は、立方体の構成を示しており、V1～V8は立方体の頂点を示し、P1～P6は立方体を構成する多角形(ここでは正方形)を示す。このような立方体は、

例えば図4に示したように、いくつかの多角形の集合で、それぞれの多角形の頂点(V1, V2など)と各頂点の法線ベクトル(n1, n2など)で与えられる。ただし、ここには図2の立方体のうちの多角形P1~P3のデータのみを示した。一方、図5は、複数の多角形に分割して与えられた曲面を示し、実線および破線で区切られた部分が分割された多角形を示し、矢印は各頂点の法線ベクトルを示す。また、図6は、ここで用いた辺属性テーブルの構成と、図2の立方体をこの辺属性テーブルに展開したものを示す。ただし、図2の立方体は図4のような図形データで与えられるとする。ここで、辺属性テーブルの、座標1、法線ベクトル1、座標2、法線ベクトル2はそれぞれ、辺の始点の座標、始点の法線ベクトル、終点の座標、終点の法線ベクトルを示し、座標1と座標2は輪郭線、稜線を検出する際の始点と終点になり、法線ベクトル1と2は稜線と近似線とを判定する際に用いる。多角形IDとは、例えば図2に示した多角形P1, P2, ...を識別するための番号であり、多角形ID1と2はその辺を境に接している多角形の識別子を入力し、この値は輪郭線、稜線の判定時に用いる。辺種別とは、多角形を構成する辺がどのように2つの多角形を接続しているかを示す。ここでは変種別を、単独の辺、多角形と多角形とを接属する稜線、近似線の3種類に分類し、それぞれ0, 1, 2とする。ここで、単独の辺とは、図5の実線で示したような多角形には含まれていない辺を、稜線とは、図2の立方体を構成するような辺で、多角形には含まれ、その辺の始点と終点のそれぞれの点で2方向の法線を有する辺を、また、近似線とは、図5の破線で示したような辺で、多角形には含まれ、その辺の始点と終点のそれぞれの点で1方向のみの法線を有する辺をそれぞれ示す。

【0018】図7は、これら図2、図4、図5に示したデータから図6の辺属性テーブルを作成する処理の流れを示している。なお、この処理は、図10におけるCPU1001で実行され、図4、図5の等の図形データは図形データエリア1003に、図6の辺属性テーブルは辺属性テーブルエリア1004に保存される。まず、現在判定の対象となる辺と、始点と終点の座標が一致する辺が今までに登録されているかを、判定701で調べる。一致する辺が存在しない場合は、新たな辺として追加する処理707、708を実行する。一致する辺が存在した場合には、一致した辺に関して処理702を実行し、続いて判定703により稜線か、近似線なのかを判定し、稜線であれば処理705を、近似線(この辺は表示しない)の場合には処理704を実行する。この判定は、図4および図5に示したように、始点と終点の法線ベクトルが異なる値(図4)の場合には稜線とし、同じ値(図5)の場合には近似線とする。これらの処理を全ての面の全ての辺に関して実行すれば辺属性テーブルが完成する。

【0019】図2に示した立方体に前記の処理を施すと以下ようになる。辺V1V2に関しては座標1、座標2はそれぞれV1, V2となり、多角形ID1、多角形ID2はそれぞれP1, P2となる。また図4より、多角形P1の頂点V1の法線ベクトルn1と、多角形P2の頂点V1の法線ベクトルn1'とは方向が違い、また、多角形P1の頂点V2の法線ベクトルn2と、多角形P2の頂点V2の法線ベクトルn2'とは方向が違っているので、辺種別は1となる。一方、図5で与えられた曲面データの場合には、破線部分では辺種別は2であり、実線部分では辺種別は0となる。

【0020】ステップ102の説明: ステップ102は、Zバッファ法により、隠面消去とスムーズシェーディングが施された画像の生成であり、図形データエリア1003に保存されている図形データをCPU1001が描画命令に変換し、この描画命令をグラフィックスプロセッサ1005に送り、グラフィックスプロセッサ1005内の通常描画部1006で画像を生成し、フレームメモリ1009に色情報を記憶する。なお、Zバッファ法による多角形の表示については、前記ニューメディア技術シリーズ コンピュータグラフィックスを参照されたい。

【0021】ステップ103の説明: ステップ103は、Zバッファ法を用い、色情報の代りに各画素に表示されている多角形IDを情報とする多角形ID画像を生成する。図3は、この多角形ID画像を示しており、図2に示した立方体のある方向から見た表示例を示し、P1, P2, P3は実線で囲まれた部分の多角形IDを示す。この画像の生成は、従来の画像生成法と同じであり、図形データエリア1003に保存されている図形データをCPU1001が描画命令に変換し、この描画命令をグラフィックスプロセッサ1005に送り、グラフィックスプロセッサの通常描画部1005内の通常描画部1006で画像を生成し、色情報のかわりに多角形IDを多角形IDメモリ1010に記憶する。

【0022】ステップ104, 105の説明: 次に、ステップ104の輪郭線、稜線の検出処理、ステップ105の合成処理について説明する。それにさきだち、図10の輪郭線稜線発生部1007の詳細を図11を用いて説明する。これは、線描画部1103、周囲画素読み込み部1104、輪郭線稜線判定部1105、輪郭線稜線書き込み部1106からなる。図8は、輪郭線、稜線の検出処理の流れ図である。

【0023】まず、処理801で辺属性テーブルエリア1004内の辺iの辺属性テーブルをCPU1001が描画命令に変換し、グラフィックスプロセッサ1005内の輪郭線稜線発生部1007がこの命令を受け取る。

【0024】次に、この辺iの画素発生処理802を線描画部1103が実行する。線描画部1103は、終点と始点の座標1101を受け取り、描線アルゴリズム(DDA等)を用いて画素アドレスを発生する。そし

て、その画素位置の周囲8画素に対応する画素アドレスの多角形IDを、周囲画素読み込み部1104が多角形IDメモリ1010から読み込む。ここで、画素位置の周囲8画素とは、図3の301の拡大図9において、画素位置が901であるときに、四角形902の901以外の8画素を示す。

【0025】そして、輪郭線稜線判定部1105が、周囲画素読み込み部1104が読み込んだ周囲8画素の多角形IDと、CPU1001から受け取った辺属性テーブルの多角形ID1, 2とを比較する判定803を実行する。そして、周囲8画素のなかに辺属性テーブルの多角形ID1と2のどちらも含まない場合には、輪郭線稜線判定部1105は輪郭線稜線書き込み部1106に描画命令を出さない、すなわち、表示しない処理805を実行する。しかし、どちらか一方を含むならば輪郭線として画像に付加する処理807を実行し、両方を含む場合には辺種別の判定処理804を実行し、この値が1ならば稜線として表示する処理806を、2ならば表示しない処理805を実行するので、この場合には、輪郭線稜線判定部1105は輪郭線稜線書き込み部1106に輪郭線または稜線の描画命令を出す。

【0026】稜線の検出例として、図3の301の拡大図を図9に示す。この図において、①、②はそれぞれP1, P2の多角形IDを示し、左下から右上に引かれている直線は図2の辺V1V2に対応する。この辺属性テーブルは、図6の辺1に登録されている。いま、図6の辺1を描画しており、図9の画素901が発生されたとする。この時、発生画素の周囲8画素は、四角形902で囲まれた901を除く8画素である。この画素901は周囲の8画素に辺1の辺属性テーブルの多角形ID1, 2の両方を含み、この辺の辺種別が1なので稜線となる。

【0027】フレームメモリに、この装置を用いて検出した輪郭線、稜線のみを描画すれば、隠線消去されたワイヤフレーム画像が作成でき、また、フレームメモリにスムーズシェーディングが施された画像と、この輪郭線、稜線を他の部分と異なる色で合成して書き込めば、物体形状が理解しやすい画像が生成できる。

【0028】以上の処理を全ての辺について実施すれば、間違えなく輪郭線、稜線が検出でき、また、物体形状が理解しやすい画像が生成できる。

【0029】

【発明の効果】以上述べたように、本発明によれば、輪郭線、稜線の検出に必要なデータを作成し、輪郭線、稜線を検出するので、微分値などによりノイズを拾うこともなく、検出誤りはない。また、輪郭線、稜線は1画素

幅の直線として取り扱っているので、これらの線は太くなることはない。

【図面の簡単な説明】

【図1】本発明の処理の流れを示す図である。

【図2】立方体の構成図である。

【図3】多角形ID画像を示す図である。

【図4】与えられた立方体のデータを示す図である。

【図5】与えられた曲面のデータを示す図である。

【図6】辺属性テーブルの構成図である。

【図7】辺属性テーブルの作成処理の流れを示す図である。

【図8】輪郭線、稜線の検出処理の流れを示す図である。

【図9】301の拡大図である。

【図10】本発明の一実施例を示すシステム構成図である。

【図11】輪郭線、稜線発生部の詳細図である。

【符号の説明】

100…全体処理を示す、

101～105…本発明の処理手順概要を示す、

301…稜線となる部分の画素の並び、

701～708…辺属性テーブル作成の処理手順を示す、

801～809…輪郭線、稜線の検出処理手順を示す、

901…例として用いた画素、

902…画素901の周囲8画素を示す、

1001…CPU、

1002…主メモリ、

1003…図形データエリア、

1004…辺属性テーブルエリア、

1005…グラフィックスプロセッサ、

1006…通常描画部、

1007…輪郭線稜線発生部、

1008…Zバッファ、

1009…フレームメモリ、

1010…多角形IDメモリ、

1011…表示装置、

1012…システムバス、

1101, 1102…輪郭線稜線発生部が受けるデータ、

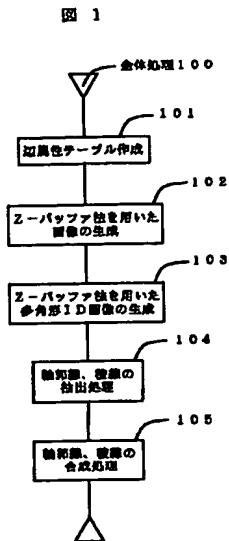
1103…線描画部、

1104…周囲画素読み込み部、

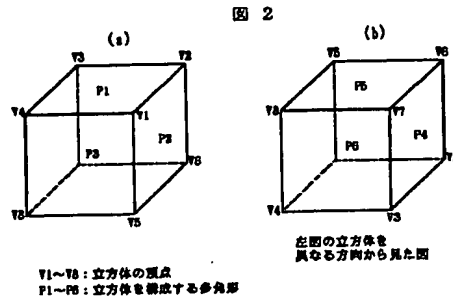
1105…輪郭線稜線判定部、

1106…輪郭線稜線書き込み部。

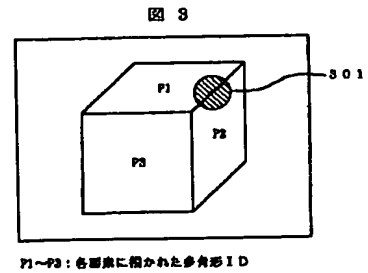
【図1】



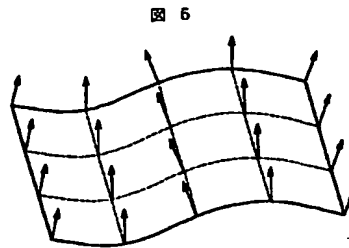
【図2】



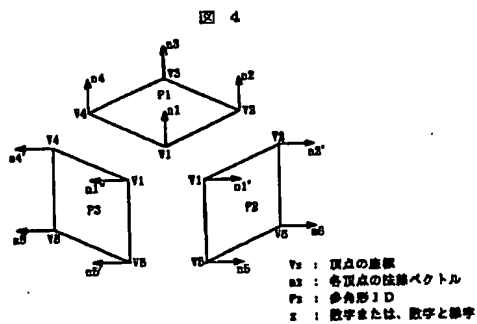
【図3】



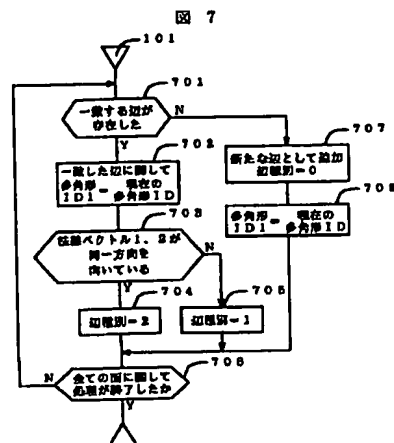
【図5】



【図4】



【図7】



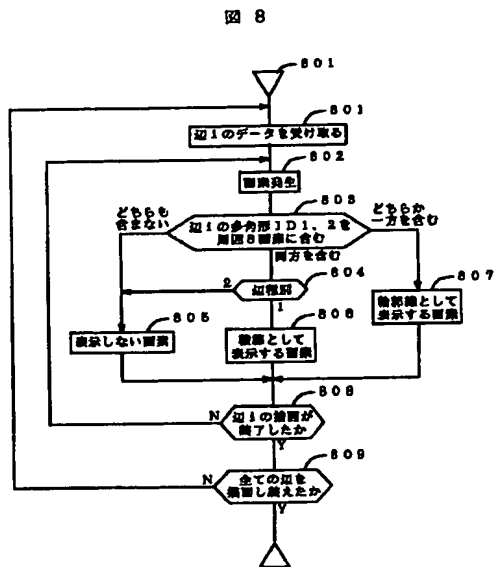
【図6】

図 6

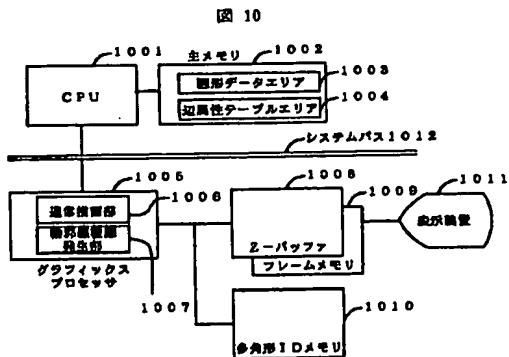
面画ID	頂点座標	法線ベクトル	多角形ID	辺属性	面属性
1	(0,0,0)	(0,0,1)	1	0	1
2	(1,0,0)	(0,0,1)	2	0	1
3	(1,1,0)	(0,0,1)	3	0	1
4	(0,1,0)	(0,0,1)	4	0	1
5	(0,0,1)	(0,1,0)	5	0	1
6	(1,0,1)	(0,1,0)	6	0	1
7	(1,1,1)	(0,1,0)	7	0	1
8	(0,1,1)	(0,1,0)	8	0	1

辺属性: 0-多角形と多角形との接続していない(単独の輪郭線)、  
1-多角形と多角形との接続であり、境界線、  
2-平面または曲面を分割する辺である(近接線)。

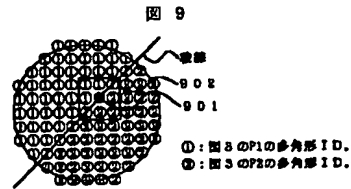
【図8】



【図10】



【図9】



【図11】

